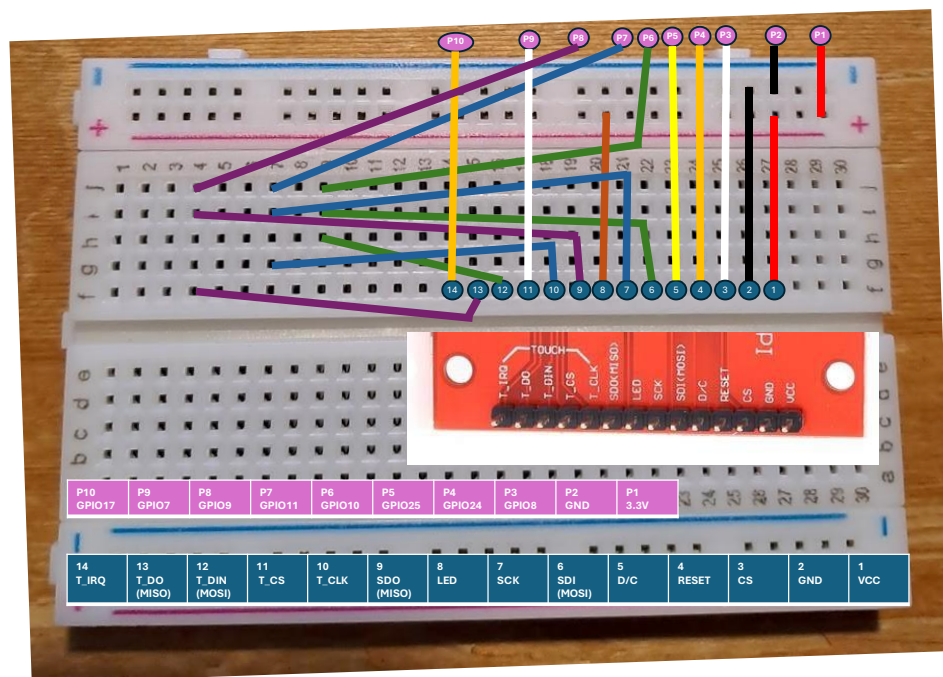


Bonjour,

1) J'ai réussi à intégrer l'écran avec le robot. Il reste des instabilités

- Pour que l'écran puisse fonctionner en même temps que les anneaux de leds j'ai proposé un nouveau câblage qui libère le GPIO18 et que l'on partage MISO, CLK et MOSI. Par contre il faudra retoucher l'UI de l'écran (touch vs touchv2 qui a été proposé pour gérer les « touch »)
- Je n'ai pas réussi (pas essayé plus que cela) à faire le code des écrans pour que les boutons soient cliquables. J'ai pour l'instant fait une grosse zone cliquable sur la home qui lance un nouveau screen le « pipeline ». Cet écran lit les événements remontés par le robot → J'ai envoyé une vidéo cela fonctionne.
  - J'ai modifié le driver touchv2 mais si vous avez plus simple → Super
  - Le seul but est de faire un écran pipeline qui affiche les remontées des états.



		Broche écran module ILI9341 2.8" sur PCB rouge, avec le connecteur 14 pins	Broche Raspberry Pi 4 AVANT	Broche Raspberry Pi 4 APRES	Couleur	Rôle
P1	1	VCC	3.3V	3.3V	Rouge	Alim 3.3V
P2	2	GND	GND	GND	Noir	Masse
P3	3	CS	GPIO8	GPIO8	Blanc	Chip Select LCD
P4	4	RESET	GPIO24	GPIO24	Orange	Reset LCD
P5	5	DC	GPIO25	GPIO25	Jaune	Data/Command LCD
P6	6	SDI (MOSI)	GPIO10	GPIO10	Vert	SPI MOSI LCD
P7	7	SCK	GPIO11	GPIO11	Bleu	SPI Clock LCD
	8	LED	3.3V	3.3V	Marron	Rétroéclairage
P8	9	SDO (MISO)	GPIO9	GPIO9	Violet	SPI MISO LCD

	10	T_CLK	GPIO18	GPIO11	Bleu	SPI Clock Touch
P9	11	T_CS	GPIO27	GPIO7	Blanc	CS Touch XPT2046
	12	T_DIN (MOSI)	GPIO23	GPIO10	Vert	MOSI Touch
	13	T_DO (MISO)	GPIO19	GPIO9	Violet	MISO Touch
P10	14	T_IRQ	GPIO17	GPIO17	Orange	Interrupt Touch (optionnel)

Pour rappel :

- Servos GPIO 23 et 24
- LEDS 18 et 19
- BOUTON SCL1(GPIO 3 / Pin5)

2) Plan pour la suite :

- Faire propre avec les câbles :
  - Câbles bien « propres »
    - Les cables des leds externes bougent – il y a des faux contacts
    - Les cables se croisent dans tous les sens
  - Rallonges qui se débranchent
  - Mettre des cable en Y pour les MISO, MOSI et CLK
  - ⇒ Dès qu'on bouge le robot il faut tout vérifier
- **On a deux choix possibles pour l'écran**
  - Soit on continue avec le cablage de base de l'écran
    - On perd le GPIO18 et on ne met pas d'anneau externe pour la lumière (uniquement la lumière du capot)
  - Soit on fait un câblage en partageant MOSI, MISO, CLK
    - Il faudra des adaptations dans le code de l'écran mais ne semble pas énorme (fichier touch dans hardware)
  - Pour intégrer l'écran au robot
    - Livré un état stable de l'écran avec boutons cliquables
      - Ajouter un bouton pour lancer le pipeline
    - Faire un nouveau « Screen » pour le résultat du pipeline (je l'ai fait)
      - Mettre des boutons pour arrêter, etc...
    - Intégrer le lancement du robot (voir patch)
      - Dans le main
        - Il faut mettre des imports des fichiers rbx..
        - Il faut rajouter des init dans dans \_\_init\_\_ de la classe RubikGUI
        - Rajouter des fonctions : start\_robot, stop\_robot, robot\_state , robot\_running
      - Dans Home
        - Ajouter une fonction on\_start qui lance le start\_robot du main

- Pour l'éclairage
  - Les leds du cotés fonctionnent (mais faux contacts)
  - Les leds du capot fonctionnent aussi
  - Pour les réglages il faut utiliser le fichier config.json (voir après)
  - Le principe est que pour essayer est de figer les réglages de la caméra on prend 4 photos (en 4 flips) pour savoir si c'est très exposé ou pas.
  - Plusieurs tests montrent que la lumière pose beaucoup de problèmes pour la qualité de la vision
- Pour le bouton d'arrêt d'urgence
  - Faire un script comme Alexi qui lance un shutdown

### 3) Ma proposition pour cette dernière semaine - Se concentrer sur l'essentiel :

- L'éclairage du capot
  - Rien à faire
- Le petit écran
  - Choisir le nouveau câblage
  - Finir le code des touches boutons
  - **Regarder** les suggestions de chat GPT
    - Le problème classique avec le XPT2046 sur Raspberry P sont les sources de perturbations sur ton setup qui sont particulièrement nombreuses : les servos qui génèrent du bruit sur l'alimentation, les NeoPixels sur GPIO, et le SPI hardware partagé.
    - ➔ « Un condensateur de découplage (100nF céramique + 10µF électrolytique) entre VCC et GND du XPT2046 directement sur ses broches d'alimentation fait souvent des miracles. Si tu ne l'as pas déjà, c'est la première chose à tester. »
    - Dans touch v2
      - ➔ Tu peux ajouter une moyenne sur plusieurs lectures consécutives avant de valider une position. Plutôt que de prendre la première valeur brute, tu lis 3-5 fois et tu fais la médiane — ça élimine les pics parasites sans trop impacter la réactivité.
      - ➔ Tu peux aussi augmenter le `move_threshold_px` à 5-8 pixels au lieu de 2, et ajouter un délai de confirmation : une position n'est validée que si elle est stable sur 2 lectures consécutives.
- - Le bouton d'arrêt d'urgence
    - Rajouter un shutdown comme Alexi
  - **Abandonner** l'éclairage par led lateral

- Avec le nouveau câblage on pourra toujours revenir dessus
- C'est une source d'ennui et de bruit
- Ce n'est pas beau et ce n'est pas intégré
- Et je sais que vous n'allez pas aimer, mais revenir des choses simples pour maîtriser la lumière :
  - RIEN
  - Une lumière externe (on pourra toujours jouer avec ma lampe à led si nécessaire)
  - Un parapluie
  - Jouer sur les configurations de lumière du config.json
- **La PO → Aller discuter avec le client pour montrer notre plan et nos solutions pour faire quelque chose de fonctionnel. Négocier. Ne pas se faire rajouter du boulot ; notre objectif est de finir.**